

Claims:

A listing of the claims has been included for reference. Please amend claims 1, 3, 5, 14, 23, 25, 27, 31, 33, and 35-36. Please cancel claims 2, 7-8, 15-22, 24, 29-30, 32, and 37-38. No new matter has been added.

Listing of Claims:

1. (Currently Amended) A method, comprising:

collecting a loop trip count continuously during runtime of a region of code being executed that contains a loop, **the loop trip count being collected each time the loop is entered by one or more monitoring counters instrumented in the region of code;**

calculating an average trip count continuously during runtime, wherein each average trip count comprises a sequential plurality of collected trip counts over an interval of time;

classifying each average ~~categorizing the~~ trip count to be in one of a plurality of conditions, the one of the plurality of conditions to identify one or more code modification techniques applicable to the loop;

dynamically applying, during the same runtime, the one or more applicable code modification techniques to alter the code that relates to the loop **when the average trip count classification changes;**

halting the trip count data collection during the same runtime when a threshold value of consecutive intervals of time pass without a trip count classification change; and
removing one or more monitoring counters, during the same runtime, from the region of code relating to the loop when trip count data collection is halted.

2. (Cancelled)
3. (Currently Amended) The method of claim **1** ~~2~~ further comprising executing the region of code for an introductory profiling phase time interval to establish an initial average trip count value.
4. (Original) The method of claim 3, wherein dynamically applying the one or more applicable code modification techniques further comprises applying one or more scalar transformation techniques to the loop upon receiving the initial average trip count value.
5. (Currently Amended) The method of claim 3, ~~wherein categorizing the trip count~~ further **comprising comprises:**
 - determining a low trip count threshold value and a high trip count threshold value;
 - classifying the trip count as being in a first condition if the trip count is equal to or below the low trip count threshold value;

classifying the trip count as being in a second condition if the trip count is above the low trip count threshold value and below the high trip count threshold value; and

classifying the trip count as being in a third condition if the trip count is equal to or above the high trip count threshold value.

6. (Original) The method of claim 5 further comprising:
classifying the average trip count upon completion of each time interval subsequent to the introductory profiling phase to identify one or more loop transformation techniques applicable to the loop; and
dynamically applying the one or more applicable loop transformation techniques to alter the code that relates to the loop if the trip count classification changes.

7-8. (Cancelled)

9. (Original) The method of claim 3 further comprising:
determining if the loop has a regular control flow graph; and
applying one or more scalar transformation techniques to the code relating to the loop and one or more loop transformations to the code relating to the loop upon receiving the initial trip count value if the control flow graph is regular.

10. (Original) The method of claim 3 further comprising:
determining if the loop has substantial floating-point operations; and

applying one or more scalar transformation techniques to the code relating to the loop and one or more loop transformations to the code relating to the loop upon receiving the initial trip count value if the loop has substantial floating-point operations.

11. (Original) The method of claim 6, wherein applying loop transformations to the loop based on each trip count classification further comprises:

- applying loop peeling and loop unrolling transformations to the loop if the trip count classification is in the first condition;
- applying loop unrolling and software pipelining optimizations to the loop if the trip count classification is in the second condition; and
- applying software pipelining and data prefetching optimizations to the loop if the trip count classification is in the third condition.

12. (Original) The method of claim 1, wherein collecting a loop trip count further comprises:

- collecting a trip count for the loop each time the loop is entered; and
- calculating an average trip count using a sequential plurality of collected trip counts over a determined number of iterations through the loop.

13. (Original) The method of claim 12, wherein the number of iterations through the loop is 50,000.

14. (Currently Amended) The method of claim 1 2, wherein the interval of time is equal to one second.

15-22. (Cancelled)

23. (Currently Amended) A computer machine readable medium having embodied thereon instructions, which when executed by a machine, causes the machine to perform a method comprising:

collecting a loop trip count continuously during runtime of a region of code being executed that contains a loop, the loop trip count being collected each time the loop is entered by one or more monitoring counters instrumented in the region of code;

calculating an average trip count continuously during runtime, wherein each average trip count comprises a sequential plurality of collected trip counts over an interval of time;

classifying each average ~~categorizing the~~ trip count to be in one of a plurality of conditions, the one of the plurality of conditions to identify one or more code modification techniques applicable to the loop;

dynamically applying, during the same runtime, the one or more applicable code modification techniques to alter the code that relates to the loop when the average trip count classification changes;

halting the trip count data collection during the same runtime when a threshold value of consecutive intervals of time pass without a trip count classification change; and

removing one or more monitoring counters, during the same runtime, from the region of code relating to the loop when trip count data collection is halted.

24. (Cancelled)

25. (Currently Amended) The machine readable medium of claim **23** **24**, wherein the method further comprises executing the region of code for an introductory profiling phase time interval to establish an initial average trip count value.

26. (Original) The machine readable medium of claim 25, wherein dynamically applying the one or more applicable code modification techniques further comprises applying one or more scalar transformation techniques to the loop upon receiving the initial average trip count value.

27. (Currently Amended) The machine readable medium of claim 25, **wherein categorizing the trip count further comprising comprises:**

- determining a low trip count threshold value and a high trip count threshold value;
- classifying the trip count as being in a first condition if the trip count is equal to or below the low trip count threshold value;
- classifying the trip count as being in a second condition if the trip count is above the low trip count threshold value and below the high trip count threshold value; and
- classifying the trip count as being in a third condition if the trip count is equal to or above the high trip count threshold value.

28. (Original) The machine readable medium of claim 27, wherein the method further comprises:

classifying the average trip count upon completion of each time interval subsequent to the introductory profiling phase to identify one or more loop transformation techniques applicable to the loop; and

dynamically applying the one or more applicable loop transformation techniques to alter the code that relates to the loop if the trip count classification changes.

29-30. (Cancelled)

31. (Currently Amended) A system, comprising:

a bus;

a processor coupled to the bus;

a network interface card coupled to the bus; and

memory coupled to the processor, the memory adapted for storing instructions,

which upon execution by the processor **causes the processor to**

collect ~~collects~~ a loop trip count continuously during runtime of a region of code being executed that contains a loop, **the loop trip count being collected each time the loop is entered by one or more monitoring counters instrumented in the region of code;**

calculate an average trip count continuously during runtime, wherein each average trip count comprises a sequential plurality of collected trip counts over an interval of time;

classify each average ~~categorizes the~~ trip count to be in one of a plurality of conditions, the one of the plurality of conditions to identify one or more code modification techniques applicable to the loop; ~~and~~

dynamically ~~apply~~ applies, during the same runtime, the one or more applicable code modification techniques to alter the code that relates to the loop when the average trip count classification changes;

halt the trip count data collection during the same runtime when a threshold value of consecutive intervals of time pass without a trip count classification change; and

remove one or more monitoring counters, during the same runtime, from the region of code relating to the loop when trip count data collection is halted.

32. (Cancelled)

33. (Currently Amended) The system of claim ~~31~~ 32, wherein the system executes the region of code for an introductory profiling phase time interval to establish an initial average trip count value.

34. (Original) The system of claim 33, wherein the system applies one or more scalar transformation techniques to the loop upon receiving the initial average trip count value.

35. (Currently Amended) The system of claim 33, wherein the **instructions in the memory further cause the processor to system:**

determine determines a low trip count threshold value and a high trip count threshold value;

classify ~~classifies~~ the trip count as being in a first condition if the trip count is equal to or below the low trip count threshold value;

classify ~~classifies~~ the trip count as being in a second condition if the trip count is above the low trip count threshold value and below the high trip count threshold value; and

classify ~~classifies~~ the trip count as being in a third condition if the trip count is equal to or above the high trip count threshold value.

36. (Currently Amended) The system of claim 35, wherein the **instructions in the memory further cause the processor to system:**

classify ~~classifies~~ the average trip count upon completion of each time interval subsequent to the introductory profiling phase to identify one or more loop transformation techniques applicable to the loop; and

dynamically **apply** ~~applies~~ the one or more applicable loop transformation techniques to alter the code that relates to the loop if the trip count classification changes.

37-38. (Cancelled)